EXHIBIT 1C

Exhibit C

EXPERT DISCLOSURE

MATTHEW J. EDMAN, PH.D.

Matthew J. Edman, Ph.D. is a Partner and Co-Founder of NAXO Labs LLC ("NAXO"), a blockchain investigations firm located in New York, NY. Dr. Edman routinely works with law enforcement, regulatory agencies, legal counsel, and private individuals on matters related to blockchain analysis and investigations. He may be called to testify regarding blockchain technology and transaction tracing, smart contracts, and internet applications.

I. QUALIFICATIONS & PRIOR TESTIMONY

1. Dr. Edman holds the following academic degrees: (a) Ph.D., Computer Science, Rensselaer Polytechnic Institute (2011); (b) M.S., Computer Science, Rensselaer Polytechnic Institute (2008); and (c) B.S., Computer Science, Baylor University (2005). His professional training includes the following industry-recognized certifications related to cryptocurrency investigations: Chainalysis Reactor Certification ("CRC"), Chainalysis Ethereum Investigations Certification ("CEIC"), and Chainalysis Investigation Specialist Certification ("CISC"). Dr. Edman also holds an AccessData Certified Examiner ("ACE") credential, which is a certification recognized in the field of digital forensics. A copy of Dr. Edman's *curriculum vitae* is attached as **Exhibit C**.

2. Dr. Edman has authored or co-authored multiple research papers in peer-reviewed conferences and journals related to techniques for cryptographic security and authentication in wireless networks, and to the design, implementation, and analysis of anonymous communication systems on the Internet. He has served as a member of the technical program committees for the Association for Computing Machinery's Conference on Computer and Communications Security and the International Financial Cryptography Association's International Conference on Financial Cryptography and Data Security. He has also served as an external reviewer for several academic conferences and journals, including The Institution of Engineering and Technology's Information Security Journal and the Privacy Enhancing Technologies Symposium. A summary of Dr. Edman's publications is included in Exhibit C.

3. In addition to his education and training, Dr. Edman has over a decade of professional experience conducting crypto asset and digital forensic investigations.

4. From 2009 to 2013, Dr. Edman was employed as a Lead Cyber Security Engineer at The MITRE Corporation, a federally funded research and development center, where he primarily supported the FBI's Remote Operations Unit in Quantico, VA, on various matters related to the investigation of darknet marketplaces and other illicit uses of crypto assets.

5. From 2013 to 2014, he was employed as a Senior Vulnerability Engineer by Bloomberg, LP, a global financial services, software, and media company. As a member of the firm's Vulnerability Analysis Team, Dr. Edman's responsibilities included source code reviews, design reviews, penetration testing, and "red team" analysis of the firm's internal and third-party applications, networks, and websites.

6. From 2014 to 2015, Dr. Edman was employed as a Senior Director in FTI Consulting, Inc.'s Global Risk and Investigations Practice in the Cyber Security & Investigations Group. His work included conducting crypto asset investigations and digital forensic analysis for multiple clients, including for the U.S. Attorney's Office for the Southern District of New York.

7. From 2015 to 2022, Dr. Edman was employed by Berkeley Research Group, LLC ("BRG"), a global strategic advisory and expert consulting firm, as a Director in BRG's Cyber Operations & Incident Response practice where he regularly provided expert consultation to clients regarding the investigation and analysis of addresses and transactions related to numerous different crypto assets, including Bitcoin and Ethereum.

8. Dr. Edman has previously testified as an expert in blockchain analysis and source code review in the United States District Court for the Southern District of New York, including most recently in the matter of *Securities and Exchange Commission v. Terraform Labs Pte. Ltd. and Do Hyeong Kwon* (1:23-cv-01346-JSR). A summary of his prior testimony is included in Exhibit 1.

II. <u>ANTICIPATED OPINIONS</u>

9. If called as a witness at trial, Dr. Edman may offer the following opinions based on his academic and professional education, training, and experience, as well as his review and analysis of documents and other materials, including: (i) the pleadings in this case; (ii) the disclosures submitted by the Government's proffered experts; (iii) the materials provided in connection with those disclosures; and (iv) any additional materials cited herein which Dr. Edman relied on in forming his expert opinions.

A. **Ethereum is a blockchain protocol that relies on a distributed network of node operators to relay and process transfers of value between Ethereum users**

10. Crypto assets (or "cryptocurrencies") are digital assets that exist only in electronic form (rather than as physical coins or bills) and use cryptography to create and verify transfers from one party to another. Rather than rely on a centralized entity to validate and record transfers between users, crypto asset systems, like Ethereum, rely on a distributed network of computers (or "nodes") to cooperatively relay, process, validate, and record transfers of crypto assets between users. Ethereum's standard (or "native") crypto asset is called "ether" (or "ETH").

11. Ethereum is a public, distributed network comprising nodes operated by multiple independent operators. Users of Ethereum are represented by one or more "addresses." An address is a pseudonymous identifier derived from a "private key," which is a randomly generated cryptographic secret key typically known only to the user who generated it. The private key acts like a secret password that allows a user to access and manage their crypto assets without having to disclose their password.

12. The private key allows users to digitally sign transfers of other crypto assets from one address to another (called a "transaction"), which are distributed to and validated by other Ethereum users (e.g., by ensuring that the transaction is digitally signed by the sender and that the sender is not attempting to transfer crypto assets they do not have). In other words, Ethereum

addresses are like "accounts" and transactions can be thought of generally as simply updating account balances on a distributed public ledger. All transactions on Ethereum are public.

13. In a "proof-of-work" blockchain, special nodes called "miners" group valid transactions from multiple users into a "block," and each block is cryptographically linked to the previously mined block thus forming a "blockchain." Miners attempt to solve a cryptographic puzzle based on the new block of transactions. The first miner to solve the cryptographic puzzle distributes their solution (or "proof") to the rest of the network who can verify the solution. If a majority of the network agrees the solution is valid, that miner is awarded a certain number of crypto assets, plus potentially additional "fees" paid by the users who initiated the transactions included in the block.

14. In a "proof-of-stake" blockchain, special nodes called "validators" are chosen to create and digitally sign new blocks of transactions on behalf of the network without having to solve a cryptographic puzzle like in a proof-of-work blockchain. Instead, users can opt to lock up (or "stake") a certain amount of crypto assets in exchange for the ability to become a validator. Like miners in a proof-of-work blockchain, validators in a proof-of-stake blockchain earn a fee for their services if a majority of the network, including other validators, agree that the proposed new block is valid. Ethereum was originally a "proof-of-work" blockchain when it was created in 2015 and later migrated to a "proof-of-stake" protocol in September 2022.

15. Since each block is cryptographically linked to the previously mined or validated block, the Ethereum blockchain is "immutable" in that once a new block has been distributed to and accepted by the network, no changes can be made to that block or any previous block, or to any transactions within those blocks without affecting later blocks in the blockchain. Consequently, any transaction or other information published to the Ethereum blockchain is effectively permanently and publicly available.

16. Importantly, no transfer of crypto assets occurs unless and until a transaction is (a) cryptographically signed by the initiator of the transaction, (b) distributed to the Ethereum network, (c) verified by a miner in a proof-of-work blockchain or a validator in a proof-of-stake blockchain protocol, (d) included in a new block by that miner or validator, and (e) the new block containing the transaction is recognized as valid by a majority of the blockchain network. This testimony will be based on Dr. Edman's education, training and experience, and documentation and other public information regarding the Ethereum blockchain.

B. Tornado Cash is a non-custodial "DeFi" protocol comprising multiple smart contracts on the Ethereum network

17. In addition to transferring crypto assets between users, Ethereum transactions can use a special-purpose scripting system (akin to a programming language) to create and interact with blockchain-based computer programs called "smart contracts." Smart contracts have enabled software developers to build a diverse set of applications on the Ethereum blockchain and other blockchains without incurring the time and expense of creating an entirely new crypto asset protocol, developing a new software implementation, and attracting a network of miners or validators.

18. Decentralized applications (or "dApps") refers to software applications created using smart contracts on a blockchain like Ethereum for data processing and storage, rather than relying on a traditional centralized server and database like a typical web application. There are many examples of dApps that serve a variety of purposes—sometimes collectively referred to as "decentralized finance" or "DeFi" applications—including "decentralized exchanges" for trading one type of crypto asset for another (e.g., Uniswap) via automated smart contracts or for borrowing and lending crypto assets (e.g., Aave).

19. Tornado Cash is an example of a dApp, since it relies on smart contracts and a distributed network of miners or validators to facilitate and record transfers of crypto assets on a blockchain, rather than relying on a centralized server and database. The Tornado Cash smart contracts define the series of instructions (or "protocol") required to deposit or withdraw to or from the "pools" of crypto assets. The smart contracts associated with each pool define the crypto asset type (e.g., ETH) and amount (e.g., 0.1, 1, 10, or 100 ETH) users can deposit or withdraw in a transaction. The Tornado Cash protocol also defines how the various smart contracts comprising Tornado Cash interact with each other.

20. Users typically interact with Tornado Cash and other DeFi applications using what is commonly referred to as a "software wallet," which is an application running on the user's computer or mobile device responsible for creating and managing the cryptographic private keys necessary to digitally sign blockchain transactions. Examples of common software wallet applications used to interact with dApps include the proprietary Coinbase Wallet, and a popular open-source alternative called MetaMask.

21. Since the user controls the computer or mobile device on which the wallet application is installed and the private keys generated and held therein, this type of wallet is also referred to as a "self-custody" or "non-custodial" wallet. In contrast, centralized exchanges (e.g., Coinbase or Gemini) are considered "custodial wallets" since they generate and manage private keys on behalf of their users, and the user does not have access to or control over those private keys.

22. Tornado Cash, like other dApps, does not gain custody over its users' crypto assets at any point. The private keys required to deposit crypto assets into the Tornado Cash protocol are generated and controlled by the user via their self-custody wallet. Similarly, the "secret note" required to withdraw crypto assets from the Tornado Cash protocol is generated on the user's computer or mobile device and not transmitted to the Tornado Cash smart contracts or to any other smart contract, website, database, or other service. In this manner, Tornado Cash is similar to decentralized exchanges like Uniswap or OpenSea, which simply provide a platform for users to transact via automated smart contracts and do not take custody of users' crypto assets like in a centralized exchange.

23. This testimony will be based on Dr. Edman's education, training and experience, publicly available documentation about the Ethereum blockchain and related DeFi protocols, developer documentation and source code for the Tornado Cash smart contracts, and source code for the Tornado Cash UI and CLI.

C. The "Tornado Cash UI" and "Tornado Cash CLI" software created transactions on user devices which were submitted to the Ethereum network via third-party services.

24. When the Tornado Cash protocol was released in 2019, an implementation of it could be accessed via the domain "tornado.cash" using a typical web browser. The website provided access to a graphical user interface (or "UI") that performed the cryptographic operations necessary to interact with the Tornado Cash protocol, including: (a) generating a random value called a "secret note" required to deposit crypto assets into the Tornado Cash protocol, and (b) creating a "zero-knowledge proof" required to withdraw crypto assets from the Tornado Cash protocol, which demonstrates (or "proves") knowledge of the secret note but without disclosing the note itself. Those cryptographic operations occurred entirely within the user's web browser running on their computer or mobile device and did not rely on a centralized server to function.

25. The Tornado Cash UI obtained information from the Ethereum blockchain (e.g., the balance of crypto assets associated with a wallet address) via a "remote procedure call" (or "RPC") service. RPC refers to a standardized protocol that allows one computer program to interact with another computer program over a network. The UI used a variety of third-party RPC services, including a commercially available service called Infura. Users could also choose to use a different third-party RPC service or operate their own instead.

26. The Tornado Cash UI created the series of instructions (i.e., the blockchain transaction) representing deposits to or withdrawals from the Tornado Cash protocol within the user's web browser. The UI would then programmatically request the user's self-custody wallet software (e.g., MetaMask) to cryptographically sign those transactions using the private keys held on the user's device, which would typically have to be approved by the user within the self-custody wallet software.

27. In the case of deposits to the Tornado Cash protocol, the UI created the transactions that were then submitted to the Ethereum network by the user's third-party wallet software. By default, the UI would cause withdrawal transactions to be submitted to the Ethereum network via a "relayer." A relayer is a service operated by a third-party who submits (or "relays") the user's withdrawal transaction to the Ethereum blockchain and pays the required transaction fees (or "gas") in exchange for a reward paid from that transaction. This is similar to other services like 0x's "gasless API" which is a third-party service that submits transactions to the blockchain and pays the required gas fees on behalf of users in exchange for a portion of the crypto assets in the transaction. If the user chose not to use a relayer, the withdrawal transactions were transmitted to the Ethereum blockchain by the user's self-custody wallet.

28. In January 2020, the Tornado Cash UI became available via a publicly available peer-to-peer network called the InterPlanetary File System ("IPFS"). IPFS provides the ability for users to store and share data across a distributed network of nodes. Unlike normal websites which are accessed via a domain name (e.g., "tornado.cash"), files or other content made available via IPFS are accessed using a unique identifier called a "content identifier" (or "CID") based on a cryptographic hash of the data. Users could access and download the UI through IPFS software running on their computer or by using a third-party IPFS "proxy" service that was accessible with a normal web browser and acted as a relay between the user's web browser and the IPFS network.

Users could also host the Tornado Cash UI on their own IPFS node, which would make the Tornado Cash UI available via the same CID if the UI had not been modified.

29. The CID of the Tornado Cash UI could be obtained from another user or by resolving a Ethereum Name Service ("ENS") name. An ENS domain (e.g. "tornadocash.eth") is similar to a typical Internet domain name (e.g., "tornado.cash"), except an ENS name is mapped (or "resolved") to a wallet address or other data using smart contracts on the Ethereum blockchain. Some third-party IPFS proxy services, such as "eth.link" and "eth.limo," can automatically resolve ENS names to a CID and display the corresponding content. For example, accessing the domain "tornadocash.eth.link" with a typical web browser would cause the "eth.link" IPFS proxy service to resolve the ENS name "tornadocash.eth" to a CID, retrieve that CID from IPFS, and return the result to the user's web browser.

30. On or about September 16, 2021, the Tornado Cash UI was removed from the "tornado.cash" website and replaced with a link to "tornadocash.eth.link." On or about April 12, 2022, "tornadocash.eth.link" was replaced with "tornadocash.eth.limo."

31. In addition to the Tornado Cash UI, users could also interact with the Tornado Cash smart contracts via a "command-line interface" ("CLI") that users could download and run on their own computer. The CLI software was created on or about May 21, 2020, and its source code was publicly available no later than September 11, 2020. The CLI generated a secret note onto the user's computer and would not be transmitted to the Tornado Cash smart contracts or to any other smart contract, website, database, or other service. The user was then required to specify a third-party RPC service, or an RPC service operated by the user which obtained information from the Ethereum blockchain. The CLI then would transmit deposit the transactions to the Ethereum blockchain via the user-chosen RPC service, and the withdrawal transactions would be transmitted to the Ethereum blockchain via a relayer.

32. The Tornado Cash UI source code was implemented using a common web application framework called Vue.js. It was initially made publicly available in a "minified" format, which is a common technique used by web application developers for the purpose of reducing the size of the source code. Source code is often minified when "compiling" and distributing web applications that operate entirely within a user's web browser and do not rely on a centralized web server to interpret the source code and "render" it for the user's web browser as the user interacts with the application (e.g., "single page applications"). Since IPFS only provides content distribution and does not function as a traditional web server, the UI source code must be compiled for it to be accessible over IPFS.

33. Since the source code for the Tornado Cash UI and CLI software was publicly available via GitHub, the developers could not exercise control over what users did with that source code any more than the developers of other open-source applications (e.g., MetaMask) since that software runs on a user's computer or mobile device rather than on a centralized server. Software developers generally also cannot prevent users from modifying that source code or from using a previously available version once it has been released.

34. This testimony will be based on Dr. Edman's education, training and experience, as well as blockchain data, source code and developer documentation for the Tornado Cash UI and CLI, source code for the Tornado Cash smart contracts referenced above, public documentation regarding IPFS, and archived information from the GitHub website.

D. <u>The smart contracts associated with the Tornado Cash "pools" were immutable as of May 2020 and could not be modified to restrict deposits or withdrawals</u>

35. Ethereum smart contracts comprise two main components: (a) the series of instructions that specify the functionality (or "implementation") of the smart contract, and (b) data associated with the smart contract that the implementation can use to store additional information, such as account balances and other information which varies depending on the implementation of the smart contract. Due to the immutable nature of the Ethereum blockchain, the series of instructions comprising the implementation of a smart contract cannot be modified once published. Depending on how a smart contract is programmed, however, the data associated with that smart contract can be updated after it is created. It cannot, however, be used to remove the smart contract from the blockchain.

36. The Tornado Cash pool smart contracts contained data that identified a particular wallet address called the "operator" address. The implementation of the smart contracts provides the operator address the ability to update certain information required to process and verify deposits and withdrawals to and from the Tornado Cash pools. The operator address could also transfer that role to a different address. However, even the operator address could not modify the series of instructions defining the operation of the Tornado Cash pools or remove the smart contracts from the public Ethereum blockchain.

37. Typically, the address which "owns" a smart contract is initially set to the address used to create the smart contract on the blockchain. In or around May 2020, the operator address associated with the Tornado Cash pools was changed to reference a non-existent (or "null") wallet address. The "null" address, represented by an address of all zeros, is not owned by any Ethereum user. Consequently, once the operator role was changed to the "null" address, the data associated with the Tornado Cash smart contracts could no longer be modified, including the ability to transfer the operator role to a new address.

38. Given the immutability of smart contracts once they are deployed, the use of "proxy contracts" (or simply "proxies") are a common design pattern for allowing DeFi protocols to be "upgraded" or modified after they are published to the blockchain even though the individual smart contracts comprising the protocol cannot be modified. A proxy contract has associated with it the address of another contract called an "implementation" or "logic" contract. Users create transactions to interact with the proxy address, and the proxy contract simply redirects (or "proxies") requests to the implementation contract. If the developers identify a flaw in their implementation contract or add new functionality, they can deploy a new implementation contract and update the data associated with the proxy contract to relay those requests to the new implementation contract. Proxy contracts are like a mail forwarding service that receives mail at a particular address and then forwards that mail to the intended recipient. If the recipient's mailing

address changes, only the mail forwarding service needs to be notified rather than every potential sender.

39. While certain smart contracts associated with the Tornado Cash protocol utilized this upgradable "proxy" architecture, the smart contracts associated with the Tornado Cash pools did not. In other words, updating the data associated with the proxy contracts to forward transactions to a new implementation contract would not prevent anyone from interacting directly with the immutable Tornado Cash pools.

40. This testimony will be based on Ethereum developer documentation, announcements made by the Tornado Cash developers, and public blockchain data.

E. **The Tornado Cash DAO delegated governance of certain aspects of the Tornado Cash protocol to TORN token holders.**

41. A decentralized autonomous organization (or "DAO") is a governance structure commonly adopted by DeFi protocols. A DAO is typically formed using a smart contract, which defines the rules regarding how a DeFi protocol can be modified by participants in the DAO. Some DAOs require users to hold a particular crypto asset associated with the DAO called a "governance token." Governance token holders may participate in the DAO by creating and voting on proposals to update the protocol in accordance with the logic defined in the governance contract.

42. In December 2020, a new series of smart contracts establishing a governance structure for the Tornado Cash protocol was deployed. The governance contract delegated the ability to update or modify certain aspects of the Tornado Cash protocol to governance participants. Given the immutability of the Ethereum blockchain, only certain aspects of the Tornado Cash protocol could be modified by governance proposals, such as creating new Tornado Cash pools. The proxy contracts, along with other aspects of the Tornado Cash protocol including the Relayer Registry, were also controlled by vote via the Tornado Cash governance protocol. However, governance proposals could not be used to modify, remove, or disable the existing immutable Tornado Cash pools.

43. The new governance structure also involved the creation of an ERC-20 crypto asset on Ethereum called "TORN" which functioned as a governance token. Any Ethereum user who held at least 1,000 TORN tokens could submit proposals to be considered and voted on by other TORN token holders who had deposited (or "staked") TORN tokens into the governance contract. If at least 25,000 TORN tokens are used to vote on a proposal during a five-day period, and a majority of votes are in favor of the proposal, then any user can implement or "execute" the proposal after a waiting period of two days.

44. Initially, 10 million TORN tokens were created and distributed to the Tornado Cash founders, early users of Tornado Cash, and others associated with the Tornado Cash project. Each Tornado Cash founder received 822,407 TORN tokens, which were initially held in a "vesting contract" for each founder. The vesting contracts would only permit withdrawal of one-third of the TORN tokens after a one-year "cliff," with 1/24 of the remaining TORN tokens vesting every 30 days thereafter.

45. Dr. Edman will offer testimony regarding when TORN tokens associated with the vesting contract identified by the ENS name "team3.vesting.contract.tornadocash.eth" were claimed and subsequently "sold," and the extent to which such TORN was used to participate in Tornado Cash governance proposals. Dr. Edman will also offer testimony regarding proposals submitted via the governance contract, including updates to the "tornadocash.eth" ENS name and associated IPFS CID for the Tornado Cash UI.

46. This testimony will be based on public blockchain data, Ethereum developer documentation, crypto asset exchange records, public statements by the Tornado Cash developers, and source code and documentation associated with the Tornado Cash governance contracts and other smart contracts referenced above.

F. Integrating the "Chainalysis Sanctions Oracle" into the Tornado Cash Router and CLI would not have prevented deposits to the Tornado Cash protocol from sanctioned addresses

47. On or about June 30, 2020, the Tornado Cash developers incorporated a "geofencing" capability into the Tornado Cash website which would deny access to users accessing the website from certain countries, including Belarus, Cuba, Iran, Iraq, North Korea, and Syria. Later, on or about April 15, 2022, the Tornado Cash founders also integrated an address screening process into the Tornado Cash UI based on a smart contract called the "Chainalysis Sanctions Oracle."

48. When creating a transaction to deposit funds into the Tornado Cash protocol, the UI would query the Chainalysis Sanctions Oracle smart contract to determine whether the address of the user making the deposit was present on a list of sanctioned addresses maintained by the sanctions oracle. If the sanctions oracle returned a positive result indicating that the user's address was sanctioned, the UI would not proceed with the deposit transaction.

49. The Chainalysis Sanctions Oracle smart contract was created on or about March 10, 2022, and requires addresses to be added to or removed from a list of sanctioned addresses by the owner of the smart contract. At the time of the transactions identified by Special Agent Joel DeCapua as representing deposits to Tornado Cash from addresses associated with the Ronin hack, none of the addresses identified by Special Agent DeCapua had been added to the Sanctions Oracle's list of sanctioned addresses.

50. Additionally, since the Sanctions Oracle smart contract relies on a manually updated list of sanctioned addresses, threat actors can avoid restrictions by first sending crypto assets from a blocked address to a new, unblocked address before depositing crypto assets into a Tornado Cash pool or other DeFi protocol. Dr. Edman will offer testimony that the Tornado Cash deposit transactions identified by Special Agent DeCapua were initiated by addresses that had

received crypto assets from the 0x098B716B8Aaf21512996dC57EB0615e2383E2f96 address but were not on the Sanctions Oracle's list of sanctioned addresses.[1]

51. Thus, even if the smart contracts associated with the Tornado Cash pools could have been modified to incorporate this sanctions screening functionality, it would likely have been no more effective than the steps the Tornado Cash developers took to incorporate the Screening Oracle into the Tornado Cash UI. Similarly, even if the Tornado Cash proxy contracts had been updated to process deposits and withdrawals via a new implementation contract that incorporated the Sanctions Oracle screening, which would have required a proposal and vote from the Tornado Cash governance protocol, the Tornado Cash pools themselves could still be interacted with directly regardless of any restrictions incorporated into the proxy contracts.

52. While the Tornado Cash CLI could also have been updated to incorporate the same Sanctions Oracle screening functionality as the Tornado Cash UI, it would also have been similarly easy for a threat actor to simply remove those restrictions from the publicly available source code or to use a previous version of the source code without the added screening functionality. Since, as described above, transactions created via the Tornado Cash UI and CLI were generally submitted to the Ethereum blockchain via a third-party RPC service, the operators of those services could also have screened the addresses involved in those transactions and rejected any transactions involving sanctioned addresses or addresses otherwise associated with illicit activity.

53. This testimony will be based on source code for the Tornado Cash smart contracts referenced above, source code for the Tornado Cash UI and CLI, public documentation regarding the Chainalysis Sanctions Oracle smart contract, blockchain data reflecting when sanctioned addresses were added to the Chainalysis Sanctions Oracle smart contract, and blockchain transaction data produced in connection with Special Agent Joel DeCapua's expert disclosure.

G. <u>The "gas limit" and "gas ratio" analysis described by Mr. Werlau to differentiate between Tornado Cash UI and CLI transactions is speculative and inconsistently applied</u>

54. Mr. Werlau describes a proposed methodology for differentiating between transactions created by the Tornado Cash UI and the Tornado Cash CLI, which he relies on to opine that the Tornado Cash UI was used more frequently than the CLI between September 1, 2020, and August 8, 2022. Dr. Edman will offer testimony that Mr. Werlau does not provide the results of any testing or verification of his methodology, nor does he consistently apply that methodology to the identified transactions.

55. Nevertheless, if one accepts Mr. Werlau's methodology for classifying Tornado Cash UI and CLI transactions, Dr. Edman will offer testimony that Mr. Werlau's own analysis demonstrates that over 80% of the Tornado Cash deposit transactions identified by Special Agent DeCapua as being associated with Ronin hack were created using the Tornado Cash CLI instead

[1] In paragraph 8 of his report, Special Agent DeCapua indicates that he will offer testimony regarding the "flow of funds" from the 0x098B716B8Aaf21512996dC57EB0615e2383E2f96 address to the Tornado Cash pools. He has not provided such flow of funds analysis but to the extent he offers such testimony at trial, Dr. Edman will respond.

of the Tornado Cash UI. This testimony will be based on data produced by Mr. Werlau and Special Agent DeCapua in connection with their expert disclosures.

III. EXPERT'S APPROVAL AND SIGNATURE

I, Matthew J. Edman, pursuant to Federal Rule of Criminal Procedure 16(b)(1)(C)(v), approve my expert witness disclosure, which contains my background and qualifications, anticipated testimony, and the bases and reasons for my opinions as set forth above.

Dated: San Francisco, CA
March 5, 2025

Matthew J. Edman, Ph.D.